IN THE SPECIFICATION:

Please amend the Summary of the Invention section starting at page 3, line 5 and ending at page 9, line15 with the following amended paragraphs.

--It is an object of the present invention to substantially overcome, or at least ameliorate, one or more disadvantages of existing arrangements.

According to a first aspect of the invention, there is provided a method of generating instructions for a directed ~~adjacency~~ acyclic graph. The [[, said]] directed ~~adjacency~~ acyclic graph ~~comprising~~ comprises one or more parent nodes and one or more leaf nodes, each ~~of which said~~ parent node representing an operator and having branches to respective descendent nodes, and each ~~of which said~~ leaf node representing a graphic object having object edges.[[, said]] The method ~~comprising the steps of:~~ includes determining groups of one or more pixel locations, where the groups are bounded by object edges.[[;]] determining, for each [[said]] group, a portion of [[said] the directed ~~adjacency~~ acyclic graph in accordance with activities of the operators, ~~wherein~~ where the [[said]] portion of the directed ~~adjacency~~ acyclic graph is that portion which passes data up the directed ~~adjacency~~ acyclic graph[[; and]]. The method also includes generating, for each [[said]] group, instructions for the determined portion of the directed ~~adjacency~~ acyclic graph, ~~wherein~~ where operator instructions are generated for those operators of the determined portion of the directed ~~adjacency~~ acyclic graph having active branches and ~~wherein~~ where leaf instructions are generated for those graphic objects which are active at [[said]] the group of one or more pixel locations.

According to a second aspect of the invention, there is provided a method of generating instructions for an expression tree. The [[, said]] expression tree ~~having~~ has a plurality of nodes comprising one or more binary nodes and a plurality of leaf nodes, ~~wherein~~ where each [[said]] binary node ~~having~~ has a left branch to a descendent [[said] node and a right branch to [[a]] another descendent [[said]] node and ~~representing~~ represents a binary operation on [[said]] the two descendant nodes, and ~~wherein~~ where each [[said]] node represents a graphic object having object edges, with one or more [[said]] graphic objects overlapping. The ~~, each said~~ overlapping graphics objects ~~comprising~~ comprise a left node region, a common region, and a right node region[[, said]]. The method ~~comprising the steps of:~~ includes determining groups of one or more pixel locations, where the groups are bounded by object edges, and[[;]] determining, for each [[said]] group, whether the left and right branches of [[said]] the one or more binary nodes are active or inactive[[; ]]. The method also includes traversing, for each [[said]] group, [[said]] the expression tree, ~~wherein~~ where the left branch of any previously traversed [[said]] binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] left node region is required for the binary operation of [[said]] the previously traversed binary node and the left branch is active and the right branch is inactive of [[said]] the previously traversed binary node, and ~~wherein~~ where a right branch of any previously traversed binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] right node region is required for the binary operation of [[said]] the previously traversed binary node and the right branch is active and the left branch is inactive of

[[said]] the previously traversed binary node[[; and]]. The method also includes generating, for each [[said]] group, operator instructions for any [[said]] traversed binary node having active [[said]] right and left branches, and leaf value instructions for any traversed leaf node.

According to a third aspect of the invention, there is provided a method of rendering an expression tree into a raster pixel image having a plurality of scanlines and a plurality of pixel locations on each [[said]] scanline. The[[, said]] expression tree ~~having~~ has a plurality of nodes comprising one or more binary nodes and a plurality of leaf nodes, ~~wherein~~ where each [[said]] binary node ~~having~~ has a left branch to a descendent [[said]] node and a right branch to [[a]] another descendent [[said]] node and ~~representing~~ represents a binary operation on [[said]] the two descendant nodes, and ~~wherein~~ where each [[said]] node represents a graphic object having object edges, with one or more [[said]] graphic objects overlapping. The ~~, each said~~ overlapping graphics objects ~~comprising~~ comprise a left node region, a common region, and a right node region.[[, said]] The method ~~comprising the steps of:~~ includes generating a table representative of [[said]] the expression tree, ~~wherein said~~ where the table comprises a plurality of records corresponding to associated [[said]] binary nodes and leaf nodes, and each [[said]] record of ~~a said~~ an associated binary node comprises a first field indicating whether a [[said]] left region is required for operation of the operator of [[said]] the associated binary node, a second field indicating whether a right region is required for operation of the operator of [[said]] the associated binary node, a third field capable of indicating whether a [[said]] left branch of [[said]] the associated binary node is active, and a fourth field capable of indicating whether a [[said]] right branch of [[said]] the associated binary node is

active.[[;]] The method further includes determining groups of one or more pixel locations, where the groups are bounded by the object edges,[[;]] determining, for each [[said]] group, whether the left and right branches of [[said]] the one or more binary nodes are active or inactive[[;]], and updating, for each [[said]] group, [[said]] the third and fourth fields of [[said]] the table for [[said]] the determined active and inactive branches[[;]]. The method also includes traversing, for each [[said]] group, [[said]] the expression tree in accordance with [[said]] the updated table [[wherein]] where the left branch of any previously traversed [[said]] binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] left node region is required for the binary operation of [[said]] the previously traversed binary node and the left branch is active and the right branch is inactive of [[said]] the previously traversed binary node, and ~~wherein~~ where a right branch of any previously traversed binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or if a [[said]] right node region is required for the binary operation of [[said]] the previously traversed binary node and the right branch is active and the left branch is inactive of [[said]] the previously traversed binary node[[;]]. The method further includes generating, for each [[said]] group, operator instructions for any [[said]] traversed binary node having active [[said]] right and left branches, and leaf value instructions for any traversed leaf node[[;]], and executing, for each [[said]] group, corresponding [[said]] generated instructions so as to render [[said]] the image.

According to a fourth aspect of the invention, there is provided an apparatus for generating instructions for a directed ~~adjacency~~ acyclic graph. The[[, said]] directed

~~adjacency~~ acyclic graph ~~comprising~~ comprises one or more parent nodes and one or more leaf nodes, each ~~of which said~~ parent node representing an operator and having branches to respective descendent nodes, and each ~~of which said~~ leaf node representing a graphic object having object edges. [[, said]] The apparatus ~~comprising:~~ includes means for determining groups of one or more pixel locations, where the groups are bounded by object edges,[[;]] means for determining, for each [[said]] group, a portion of [[said]] the directed ~~adjacency~~ acyclic graph in accordance with activities of the operators, ~~wherein~~ where the [[said]] portion of the directed ~~adjacency~~ acyclic graph is that portion which passes data up the directed ~~adjacency~~ acyclic graph[[;]], and means for generating, for each [[said]] group, instructions for the determined portion of the directed ~~adjacency~~ acyclic graph, ~~wherein~~ where operator instructions are generated for those operators of the determined portion of the directed ~~adjacency~~ acyclic graph having active branches and ~~wherein~~ where leaf instructions are generated for those graphic objects which are active at [[said]] the group of one or more pixel locations.

According to a fifth aspect of the invention, there is provided an apparatus for generating instructions for an expression tree. The[[, said]] expression tree ~~having~~ has a plurality of nodes comprising one or more binary nodes and a plurality of leaf nodes, ~~wherein~~ where each [[said]] binary node ~~having~~ has a left branch to a descendent [[said]] node and a right branch to [[a]] another descendent [[said]] node and ~~representing~~ represents a binary operation on [[said]] the two descendant nodes, and ~~wherein~~ where each [[said]] node represents a graphic object having object edges, with one or more [[said]] graphic objects overlapping. The ~~, each said~~ overlapping graphics objects ~~comprising~~ comprise a left node region, a common region, and a right node region. The [[,

said]] apparatus ~~comprising:~~ includes means for determining groups of one or more pixel locations, where the groups are bounded by object edges, and[[;]] means for determining, for each [[said]] group, whether the left and right branches of [[said]] one or more binary nodes are active or inactive[[;]]. The apparatus also includes means for traversing, for each [[said]] group, [[said]] the expression tree, ~~wherein~~ where the left branch of any previously traversed [[said]] binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] left node region is required for the binary operation of [[said]] the previously traversed binary node and the left branch is active and the right branch is inactive of [[said]] the previously traversed binary node, and ~~wherein~~ where a right branch of any previously traversed binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] right node region is required for the binary operation of [[said]] the previously traversed binary node and the right branch is active and the left branch is inactive of [[said]] the previously traversed binary node[[;]], and means for generating, for each [[said]] group, operator instructions for any [[said]] traversed binary node having active [[said]] right and left branches, and leaf value instructions for any traversed leaf node.

According to a sixth aspect of the invention, there is provided an apparatus for rendering an expression tree into a raster pixel image having a plurality of scanlines and a plurality of pixel locations on each [[said]] scanline. The[[, said]] expression tree ~~having~~ has a plurality of nodes comprising one or more binary nodes and a plurality of leaf nodes, ~~wherein~~ where each [[said]] binary node ~~having~~ has a left branch to a descendent [[said]]

node and a right branch to [[a]] another descendent [[said]] node and ~~representing~~ represents a binary operation on [[said]] the two descendant nodes, and ~~wherein~~ where each [[said]] node represents a graphic object having object edges, with one or more [[said]] graphic objects overlapping. The ~~, each said~~ overlapping graphics objects ~~comprising~~ comprise a left node region, a common region, and a right node region. The[[, said]] apparatus ~~comprising:~~ includes means for generating a table representative of [[said]] the expression tree, ~~wherein said~~ where the table comprises a plurality of records corresponding to associated [[said]] binary nodes and leaf nodes, and each [[said]] record of [[a said]] an associated binary node comprises a first field indicating whether a [[said]] left region is required for operation of the operator of [[said]] the associated binary node, a second field indicating whether a right region is required for operation of the operator of [[said]] the associated binary node, a third field capable of indicating whether a [[said]] left branch of [[said]] the associated binary node is active, and a fourth field capable of indicating whether a [[said]] right branch of [[said]] the associated binary node is active[[;]]. The apparatus also includes means for determining groups of one or more pixel locations, where the groups are bounded by the object edges.[[;]] means for determining, for each [[said]] group, whether the left and right branches of [[said]] the one or more binary nodes are active or inactive[[;]], and means for updating, for each [[said]] group, [[said]] the third and fourth fields of [[said]] the table for [[said]] the determined active and inactive branches[[;]]. The apparatus further includes means for traversing, for each [[said]] group, [[said]] the expression tree in accordance with [[said]] the updated table ~~wherein~~ where the left branch of any previously traversed [[said]] binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the

previously traversed binary node are active or [[if]] a [[said]] left node region is required for the binary operation of [[said]] the previously traversed binary node and the left branch is active and the right branch is inactive of [[said]] the previously traversed binary node, and ~~wherein~~ where a right branch of any previously traversed binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] right node region is required for the binary operation of [[said]] the previously traversed binary node and the right branch is active and the left branch is inactive of [[said]] the previously traversed binary node[[;]]. The apparatus further includes means for generating, for each [[said]] group, operator instructions for any [[said]] traversed binary node having active [[said]] right and left branches, and leaf value instructions for any traversed leaf node[[;]], and means for executing, for each [[said]] group, corresponding [[said]] generated instructions so as to render [[said]] the image.

According to a seventh aspect of the invention, there is provided a computer readable medium comprising a computer program for generating instructions for a directed ~~adjacency~~ acyclic graph. The [[, said]] directed ~~adjacency~~ acyclic graph ~~comprising~~ comprises one or more parent nodes and one or more leaf nodes, each ~~of which said~~ parent node representing an operator and having branches to respective descendent nodes, and each ~~of which said~~ leaf node representing a graphic object having object edges. The [[, said]] computer program ~~comprising:~~ includes code for determining groups of one or more pixel locations, where the groups are bounded by the object edges,[[;]] code for determining, for each [[said]] group, a portion of [[said]] the directed ~~adjacency~~ acyclic graph in accordance with activities of the operators, ~~wherein~~ where the [[said]] portion of

the directed ~~adjacency~~ acyclic graph is that portion which passes data up the directed adjacency graph[[;]], and code for generating, for each [[said]] group, instructions for the determined portion of the directed ~~adjacency~~ acyclic graph, ~~wherein~~ where operator instructions are generated for those operators of the determined portion of the directed ~~adjacency~~ acyclic graph having active branches and ~~wherein~~ where leaf instructions are generated for those graphic objects which are active at [[said]] the group of one or more pixel locations.

According to an eighth aspect of the invention, there is provided a computer readable medium comprising a computer program for generating instructions for an expression tree. The [[, said]] expression tree ~~having~~ has a plurality of nodes comprising one or more binary nodes and a plurality of leaf nodes, ~~wherein~~ where each [[said]] binary node ~~having~~ has a left branch to a descendent [[said] node and a right branch to [[a]] another descendent [[said]] node and ~~representing~~ represents a binary operation on [[said]] the two descendant nodes, and ~~wherein~~ where each [[said]] node represents a graphic object having object edges, with one or more [[said]] graphic objects overlapping. The ; ~~each said~~ overlapping graphics objects ~~comprising~~ comprise a left node region, a common region, and a right node region. The[[, said]] computer program ~~comprising:~~ includes code for determining groups of one or more pixel locations, where the groups are bounded by the object edges, and[[;]] code for determining, for each [[said]] group, whether the left and right branches of [[said]] one or more binary nodes are active or inactive[[;]]. The computer program includes code for traversing, for each [[said]] group, [[said]] the expression tree, ~~wherein~~ where the left branch of any previously traversed [[said]] binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of

[[said]] the previously traversed binary node are active or [[if]] a [[said]] left node region is required for the binary operation of [[said]] the previously traversed binary node and the left branch is active and the right branch is inactive of [[said]] the previously traversed binary node, and ~~wherein~~ where a right branch of any previously traversed binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] right node region is required for the binary operation of [[said]] the previously traversed binary node and the right branch is active and the left branch is inactive of [[said]] the previously traversed binary node[[;]], and code for generating, for each [[said]] group, operator instructions for any [[said]] traversed binary node having active [[said]] right and left branches, and leaf value instructions for any traversed leaf node.

According to a ninth aspect of the invention, there is provided a computer readable medium comprising a computer program for rendering an expression tree into a raster pixel image having a plurality of scanlines and a plurality of pixel locations on each [[said]] scanline. The[[, said]] expression tree ~~having~~ has a plurality of nodes comprising one or more binary nodes and a plurality of leaf nodes, ~~wherein~~ where each [[said]] binary node ~~having~~ has a left branch to a descendent [[said]] node and a right branch to [[a]] another descendent [[said]] node and ~~representing~~ represents a binary operation on [[said]] the two descendant nodes, and ~~wherein~~ where each [[said]] node represents a graphic object having object edges, with one or more [[said]] graphic objects overlapping. The ; ~~each said~~ overlapping graphics objects ~~comprising~~ comprise a left node region, a common region, and a right node region. The[[, said]] computer program ~~comprising:~~ includes code for generating a table representative of [[said]] the expression tree, ~~wherein said~~ where the

table comprises a plurality of records corresponding to associated [[said]] binary nodes and leaf nodes, and each [[said]] record of ~~a said~~ an associated binary node comprises a first field indicating whether a [[said]] left region is required for operation of the operator of [[said]] the associated binary node, a second field indicating whether a right region is required for operation of the operator of [[said]] the associated binary node, a third field capable of indicating whether a [[said]] left branch of [[said]] the associated binary node is active, and a fourth field capable of indicating whether a [[said]] right branch of [[said]] the associated binary node is active[[;]]. The computer program further includes code for determining groups of one or more pixel locations, where the groups are bounded by the object edges,[[;]] code for determining, for each [[said]] group, whether the left and right branches of [[said]] the one or more binary nodes are active or inactive[[;]], and code for updating, for each [[said]] group, [[said]] the third and fourth fields of [[said]] the table for [[said]] the determined active and inactive branches[[;]]. The computer program also includes code for traversing, for each [[said]] group, [[said]] the expression tree in accordance with [[said]] the updated table wherein the left branch of any previously traversed [[said]] binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] left node region is required for the binary operation of [[said]] the previously traversed binary node and the left branch is active and the right branch is inactive of [[said]] the previously traversed binary node, and ~~wherein~~ where a right branch of any previously traversed binary node is ~~traversed to its said descendent node if~~ ignored unless the right and left branches of [[said]] the previously traversed binary node are active or [[if]] a [[said]] right node region is required for the binary operation of [[said]] the

previously traversed binary node and the right branch is active and the left branch is inactive of [[said]] the previously traversed binary node[[;]], code for generating, for each [[said]] group, operator instructions for any [[said]] traversed binary node having active [[said]] right and left branches, and leaf value instructions for any traversed leaf node[[;]], and code for executing, for each [[said]] group, corresponding [[said]] generated instructions so as to render [[said]] the image.--

Please replace the paragraphs at page 10, lines 14-17, with the following amended paragraphs.

--Figs. 18A and 18B show a simple ~~compsoting~~ compositing expression illustrated as an expression tree and a corresponding description;

Fig. 19 shows an ~~exemplaty~~ exemplary expression tree for implementing a series of Porter and Duff compositing operations on objects A,B,C and D and a corresponding instruction list;--

Please replace the paragraph at page 11, lines 28-33, with the following amended paragraph.

--The general principles of the invention have application in generating instructions for directed ~~adjacency~~ acyclic graphs, and specifically expression trees. This is ~~realised~~ realized in the preferred embodiment in an activity determination and instruction generation module 500 (Fig. 5) of the pixel sequential rendering apparatus 20. This module is described in more detail in the section herein entitled "*3.0 Activity Determination and Instruction Generation Module*".--

Please replace the paragraph at page 40, lines 15-23, with the following amended paragraph.

--The Activity Determination and Instruction Generation Module 500 describe above is based on an expression tree. However, the principles of the Module 500 may be ~~generalised~~ generalized to DAGs (Directed ~~Adjacency~~ Acyclic Graphs). This can be achieved by a further embodiment of the Module 500 by allowing the parent node field of a record of the Level Activation Table to contain a list of table entries to its parent nodes, and providing an L index pointer. Changing the state of a node would then require that all of its parent nodes be modified, and instruction generation would be required to follow the L index, rather than simply looking for the next table entry. DAGs would be useful for use with clipping objects, where multiple objects are clipped by the same object.--